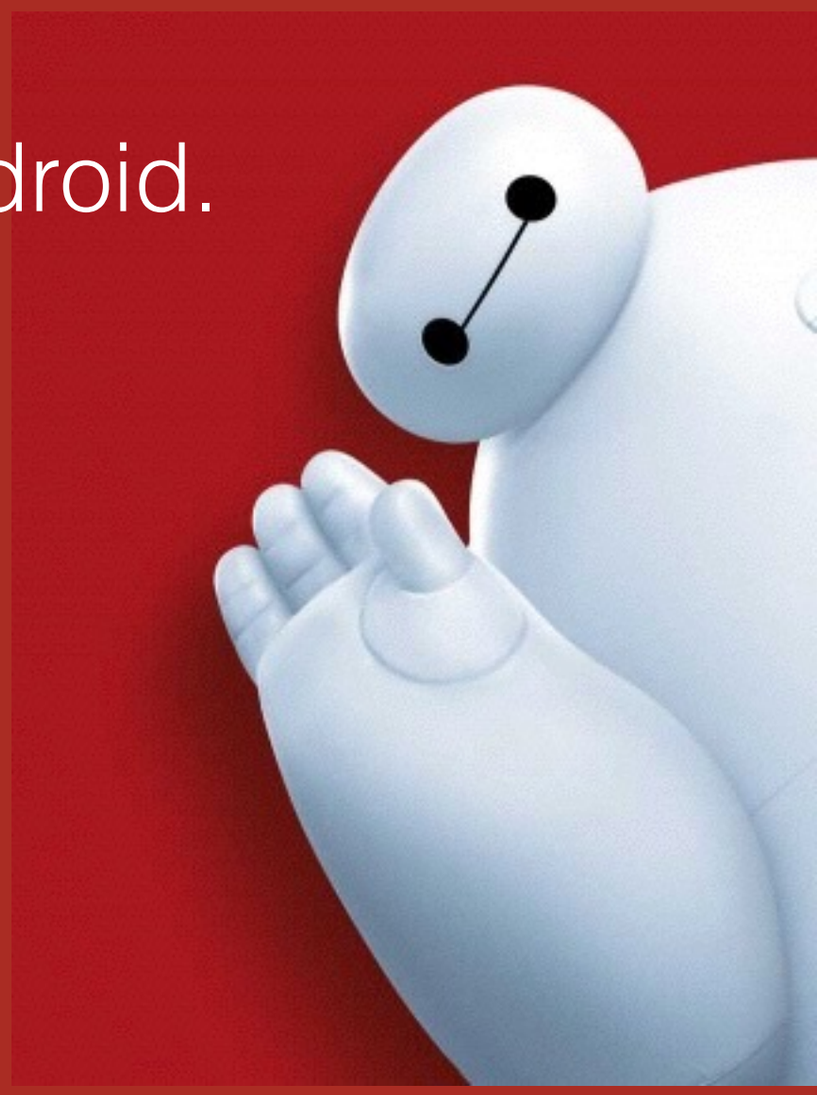
About me

- Nguyen Hoang Anh.
- Developing for 4 years now.
- Just recently made a switch to Android.



Our Android stack

- Dagger 1
- ButterKnife
- RxJava & RxAndroid
- Retrofit
- Picasso
- Realm.io / GreenDAO

Dagger 1.x

- Dependency injection framework.
- Developed by Square.
- Been in production for more than 2 years.
- http://square.github.io/dagger/

Dagger 1.x

- Code generation, partially reflection.
- Has some problems with obfuscation.

Dagger 2

- Forked from Dagger 1, developed by Google.
- Fully code generation, no reflection.
- Obfuscation works.
- We will make our transition to Dagger 2 soon.

ButterKnife

- Views injection library.
- Developed by @JakeWharton.
- http://jakewharton.github.io/butterknife/

ButterKnife

```
class ExampleActivity extends Activity {
  @Bind(R.id.title) TextView title;
  @Bind(R.id.subtitle) TextView subtitle;
  @Bind(R.id.footer) TextView footer;

  @Override public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.simple_activity);
    ButterKnife.bind(this);
    // TODO Use fields...
  }
}
```

RxJava 

- A JVM-based Reactive Extension.
- Programming against asynchronous observables.
- https://github.com/ReactiveX/RxJava

RxJava building blocks

- Observables: all the heavy works.
- Subscribers: consume the data.
- Operators: map(), flatMap(), filter() …
- Schedulers.

RxAndroid

- Provides AndroidSchedulers.mainThread()
- https://github.com/ReactiveX/RxAndroid

Retrolambda

- Because we like this fancy arrow ->
- Joking, it's really unbearable working with RxJava without Lambda.
- https://github.com/evant/gradle-retrolambda

Example

- Chaining observables: flatMap()

```
Observable<Boolean> registerObservable = // api call to register
Observable<String> loginObservable = // api call to login

registerObservable.flatMap(registerSuccessful -> loginObservable)
                  .subscribeOn(Schedulers.io())
                  .observeOn(AndroidSchedulers.mainThread())
                  .subscribe(accessToken -> {
                    // put access token to cache storage
                  });
```

Example

- Convert data to another type: map()

```
Observable<List<ProductNetworkModel>> getProductsFromNetwork = //
getProductsFromNetwork.map(products ->
                                mapToProductUiModel(products))
                      .subscribeOn(Schedulers.io())
                      .observeOn(AndroidSchedulers.mainThread())
                      .subscribe(products -> {
                        // display products on UI
                      });
```

Retrofit

- Developed by Square
- Version 1.9.0
- Square is rolling out version 2.0
- http://square.github.io/retrofit/

Retrofit 1.9.0

- Turn your REST API into a Java interface.
- Add request information using Annotation.
- Support rx.Observables.

Retrofit

```
public interface GitHubService {
  @GET('/users/{user}/repos')
  List<Repo> listRep(@Path("user") String user);

  @GET('/users/{user}/repos')
  void listRep(@Path("user") String user, Callback listRepCallback);

  @GET('/users/{user}/repos')
  Observable<List<Repo>> listRep(@Path("user") String user);
}
```

Retrofit

```
RestAdapter restAdapter = new RestAdapter.Builder()
    .setEndpoint("https://api.github.com")
    .build();

GitHubService gitHubService = restAdapter.create(GitHubService.class);

List<Repo> listRepo = gitHubService.listRep("octocat");
```

Picasso

- Image loader made easy.
- Easy to use API.
- Developed by Square.
- http://square.github.io/picasso/

Picasso

```
Picasso.with(context)
        .load("http://i.imgur.com/DvpvklR.png")
        .into(imageView);
```

Database

- Realm.io
- GreenDAO
- Both are ORMs but GreenDAO current supports unit testing a lot better.

Unit tests

- Robolectric
- Mockito
- JUnit 4

Summary

- We do have day job.
- Make use of what already there.
- Proven libraries.

Q&A